REMARKS

Applicants respectfully request further examination and reconsideration in view of the comments set forth fully below. Claims 1-15, 19-35, and 44-51 were previously pending. Within the Office Action, Claims 1-15, 19-35 and 44-51 were rejected. By the above amendment, Claims 1, 19, 24, 30, 44, 50 and 51 have been amended and new Claims 52-54 have been added. Claims 1-15, 19-35 and 44-54 are now pending.

The Applicants and their attorneys would like to thank Examiner Filipczyk and Examiner Coby for their time and courteousness during the telephonic interview on Wednesday, January 18, 2006. During the interview, differences between the present claims and U.S. Patent No. 6,438,604 issued to Kuver et al. ("Kuver"), as well as differences between the present claims and the IEEE 1394 standard protocol, which is discussed in the Background section of the present application, were discussed. No agreement was reached during this telephonic interview.

Rejections Under 35 U.S.C. § 102

Within the Office Action, Claims 1, 8, 19, 24, 30, 44, 50 and 51 have been rejected under 35 U.S.C. 102(e) as being anticipated by Kuver. The Applicants respectfully disagree with this rejection.

Kuver discloses a digital video network interface for transferring isochronous video data over an asynchronous local area network. It has become clear during the previous Office Actions and during the telephonic interview that the context of Kuver is not fully appreciated by the Examiner. This context of Kuver is very important to an appropriate understanding of what is being taught by Kuver. Kuver discusses a system, illustrated in Figure 2, that

> represents a system in which isochronous digital video data originating in a [sic] transmitting digital video (DV) camera is sent by a transmitting network interface across an asynchronous local area network, such as a Gigabit Ethernet network, to a receiving network interface which decodes and outputs the digital video data in isochronous mode to a receiving DV camera. [Kuver, col. 6, lines 47-53, Figure 2]

Kuver teaches a **transmitting-side** 1394 network interface 4, which is "a system for receiving digital video packets from the 1394 serial cable 2, *removing the 1394 data packet*

headers, repackaging two or more (preferably three) data packets into network protocol format and transmitting the data over the asynchronous network 5." [Kuver, col. 7, lines 22-27, emphasis added] Kuver also teaches a **receiving-side** 1394 network interface which "receives and unpackages network data packets arriving from asynchronous network 5, reformats the video data into IEEE 1394 format and transfers the video data packets via the IEEE 1394 serial cable 7 to the isochronous receiving unit, DV camera 8." [Kuver, col. 7, lines 37-46] Kuver further teaches that

> [u]pon ***receiving*** the IEEE 1394 data packet, physical layer 15 transmits the data packet to link layer 16. Link layer 16 interprets the data in the data packet and removes all except data, meaning that *header information, header_CRC information and data_CRC information, **are all removed***. This leaves just the data field from the packet, which DMA 17 transmits to SDRAM 22. That is, link layer 16 interprets the IEEE 1394 header and the information regarding the data in the data packet in order to know where the data came from and where the data is going. **Link layer 16 then strips off unneeded information leaving only the data field.** [Kuver, col. 9, line 61 - col. 10, line 6, emphasis added]

Kuver therefore teaches that when ***receiving*** a packet, the header and all information other than the data is stripped from the packet. In contrast to the teachings of Kuver, as will be discussed in detail below, in the present invention, when ***receiving*** a packet, the packet header is maintained with the packet and a meta data header is added to the received packet.

Within the Office Action, the concepts of reception and transmission of packets are still confused. Within the rejections and the application of the cited references, there is still a fundamental confusion regarding the treatment of packets being received and the treatment of packets being transmitted. All of the citations provided from Kuver within the Office Action are concerned with ***transmission*** of packets, not ***reception***. Specifically, in the cited section of Kuver at column 12, lines 52-59, ***transmission*** of a packet from the interface out across the IEEE 1394 bus is being taught. In this cited section it is taught that

> DMA 17 accesses the SDRAM at the location of the empty pointer, and gives the digital video data to link layer 16 for reconstruction of the headers, shown in FIG.

> 3B. After reconstruction of the IEEE 1394 headers, link layer 16 gives the data, which is now formatted in accordance with IEEE 1394 protocol to physical layer 15 for ***isochronous transmission*** out across 1394 bus to a receiving digital video device. [Kuver, col. 12, lines 52-59, emphasis added]

In this section, Kuver is teaching adding an IEEE 1394 header to a packet of data and then **transmitting** the packet across the 1394 bus. This added IEEE 1394 header is the packet header necessary for ***transmission*** across the IEEE 1394 bus. Kuver does not teach adding a meta data header to a ***received*** packet of data which already included a packet header. Further, the cited Claim 14 of Kuver reads as follows

> 14. The method according to claim 13, wherein the step of converting isochronous digital video data includes ***removing IEEE 1394 header information*** from the digital video data, ***adding network header information*** to the digital video data, repackaging the digital video data with the network header into a network packet which is formatted in accordance with the local area network protocol format and storing the network packet for ***asynchronous output in the transmitting step***. [Kuver, col. 17, lines 38-46, emphasis added]

In this cited claim, Kuver teaches stripping the IEEE 1394 header information, adding network header information and then transmitting a packet over the network. Thus, Kuver teaches stripping the IEEE 1394 packet header and then adding the network header. Kuver does not teach adding a meta data header to a ***received*** packet, which already included a packet header. Again, this cited section of Kuver is being taken out of context. Claim 14 of Kuver is dependent on the independent Claim 5. The preamble of Claim 5 reads

> A method for transmitting digital video data, for use in a digital video conferencing system, in which digital video data is transmitted from one isochronous bus to another isochronous bus via an asynchronous bus, the method comprising the steps of: [Kuver, col. 16, lines 57-61]

As evident from the teachings of Kuver and this claim, the data is transmitted over several buses, each requiring an appropriate header. But, as taught within Kuver, and discussed in detail above, at each receiving device between the buses, the received header is removed, the data is

repackaged, and a header, appropriate for the next bus, is added to the data, before the packet is transmitted.

Within the Response to Arguments section of the Office Action, it is stated that "Kuver extends the packet of data by the added network header to the received digital video data (col. 17, line 41) and stores the extended packet of data in a transmitting step (col. 17, lines 45 and 46)." [Office Action, page 6] The applicants respectfully disagree. As discussed above, in column 17, lines 38-46 (Claim 14), Kuver does not teach extending a packet of data by adding a network header. Kuver first teaches "removing IEEE 1394 header information" and then "adding network header information." **Both headers are not in the packet at the same time.** Therefore, Kuver does not teach extending the packet of data by adding a meta data header. Kuver teaches stripping one header, specific to one network (network the packet is received from), and then adding another header, specific to the other network (network the packet is transmitted on). Kuver does not teach adding a meta data header to a packet which still includes a packet header.

Furthermore, within the Response to Arguments section of the Office Action, it is stated that "Kuver does not need to remove all headers, only the unneeded headers are stripped (col. 12, lines 23-26)." The applicants respectfully disagree. The Examiner has quoted only part of the sentence. The entire sentence that the Examiner quoted in the Office Action reads, "[T]hus, step S445 bypasses the protocol stack normally associated with network controller 23, and, after stripping off unneeded network headers and the like, transfers **only the data** directly to SDRAM 22." [Kuver, col. 12, lines 23-26, emphasis added] First of all, Kuver teaches stripping off unneeded network headers and the like, meaning more than just unneeded network headers are stripped off, otherwise the sentence would not include "and the like." Secondly, Kuver finishes the sentence with "transfers only the data directly to SDRAM 22." Only the data is transferred, nothing more. Hence no headers are transferred, so they must have been stripped off. The modifier "unneeded" is referring to all network headers, which are unneeded and is not referring to some network headers, as read by the Examiner. Otherwise, more than "only the data" would be transferred. This is clearly not what is taught by Kuver. The Applicants' interpretation of this sentence is also consistent with other parts of Kuver, such as the teachings that the "Link layer 16 interprets the data in the data packet and removes all except data, meaning that header

information, header_CRC information and data_CRC information, are all removed. This leaves just the data field from the packet, which DMA 17 transmits to SDRAM 22." [Kuver, col. 9, lines 62-67]

In contrast to the teachings of Kuver, the apparatus and method of the present invention ***receives*** a received packet of data to be written to the media storage device, adds a meta data header to the received packet of data thereby forming an extended packet of data, and stores the extended packet of data onto a media within the media storage device. **The extended packet of data includes the packet header and the meta data header.** In one embodiment, referring to Figs. 4A and 4B, a series of source packets 60-63 is generated at a source device 50. The source device 50 then applies source packet headers 68-71 to each of the source packets 60-63, respectively. The source device 50 then splits the combination source packets and source packet headers into data blocks, with each source packet being split into multiple data blocks. Some number of the data blocks are then combined into an isochronous packet and the isochronous header and the common isochronous packet (CIP) header are then applied to the isochronous packet by the source device 50. Once the isochronous and CIP headers are applied to the isochronous data packet, the packet is then transmitted by the source device 50 over the IEEE 1394-1995 serial bus to the media storage device 40 of the present invention. **When the packet is received by the media storage device 40, a meta-data header is added by the media storage device 40 to the received packet**. As mentioned above, Kuver does not teach or disclose **adding a header** to the **received** packet of data. Kuver teaches stripping one header, specific to one network (network the packet is received from), and then adding another header, specific to the other network (network the packet is transmitted on).

The independent Claim 1 is directed to a method of writing data to a media storage device. The method of Claim 1 comprises receiving a received packet of data to be written to the media storage device, the received packet of data including a packet header, adding a meta data header to the received packet of data thereby forming an extended packet of data including both the packet header and the meta data header, and storing the extended packet of data onto a media within the media storage device. As described above, Kuver does not teach adding a meta header to the ***received*** packet of data thereby forming an extended packet of data including both the

packet header and the meta data header. Kuver teaches stripping one header, specific to one network (network the packet is received from), and then adding another header, specific to the other network (network the packet is transmitted on). For at least these reasons, the independent Claim 1 is allowable over the teachings of Kuver.

The independent Claim 8 is directed to a method of reading data from a media storage device which has previously been stored with header data generated by the media storage device. The method of Claim 8 comprises locating a first header data, including a cycle mark value having a pattern, reading a previously stored packet of data following the first header data from a media within the media storage device, the previously stored packet of data including a packet header, stripping the first header data from the previously stored packet of data thereby forming a retrieved packet of data, and transmitting the retrieved packet of data to another device. As described above, Kuver does not teach ***reading*** data from a media storage device by locating a first header data and reading a **previously stored** packet of data following the first header data from a media within the media storage device, the previously stored packet of data including a packet header. Kuver also does not teach stripping the first header data from a previously stored packet including a packet header and then ***transmitting*** the retrieved packet of data to another device. Kuver teaches stripping one header, specific to one network (network the packet is received from), and then adding another header, specific to the other network (network the packet is transmitted on). For at least these reasons, the independent Claim 8 is allowable over the teachings of Kuver.

The independent Claim 19 is directed to a meta data header added to received packets by a media storage device as the packets are recorded on storage media within the media storage device, each of the received packets including an existing header to which the meta data header is added such that the received packets include both an existing header and a meta data header. The meta data header of Claim 19 comprises a cycle mark value including a pattern used to locate cycle boundaries within the received packets and a cycle count value specifying a cycle number of a cycle in which the received packets are received. As described above, Kuver does not teach a meta data header added to a ***received*** packet including an existing header. Kuver teaches stripping one header, specific to one network (network the packet is received from), and then

adding another header, specific to the other network (network the packet is transmitted on). Further, Kuver does not teach a meta data header with a cycle mark value and a cycle count value. For at least these reasons, the independent Claim 19 is allowable over the teachings of Kuver.

The independent Claim 24 is directed to a media storage device. The media storage device of Claim 24 comprises means for interfacing configured for receiving a stream of data, thereby forming a received stream of data, and also for transmitting a retrieved stream of data, the received stream of data including packet header data, means for storing data for storing and retrieving the received stream of data, and means for processing coupled to the means for interfacing and to the means for storing for adding meta header data to the received stream of data as the received stream of data is received, such that each packet within the received stream of data includes both packet header data and meta header data, and providing the meta header data and the received stream of data to the means for storing for recording thereby forming a recorded stream of data, the meta header data including a cycle mark value marking cycle boundaries within the recorded stream of data. As described above, Kuver does not teach adding meta header data to the ***received*** stream of data, the received stream of data including packet header data, as the received stream of data is received and storing the header data and the received stream of data. For at least these reasons, the independent Claim 24 is allowable over the teachings of Kuver.

The independent Claim 30 is directed to a media storage device. The media storage device of Claim 30 comprises an interface circuit configured to receive a stream of data, thereby forming a received stream of data, and also to transmit a retrieved stream of data, the received stream of data including packet header data, storage media configured to store and retrieve the received stream of data, and an embedded stream processor coupled to the interface circuit and to the storage media to add meta header data to the received stream of data as it is received, such that each packet within the received stream of data includes both packet header data and meta header data, and provide the meta header data and the received stream of data to the storage media for recording to form a recorded stream of data, the meta header data including a cycle mark value marking cycle boundaries within the recorded stream of data. As described above,

Kuver does not teach an embedded stream processor to add meta header data to the ***received*** stream of data, which includes packet header data as it is received. Further, Kuver does not teach providing the meta header data and the received stream of data to the storage media for recording. For at least these reasons, the independent Claim 30 is allowable over the teachings of Kuver.

The independent Claim 44 is directed to a method of writing data to a media storage device. The method of Claim 44 comprises receiving a received packet of data to be written to the media storage device, the received packet of data including a packet header, adding a meta header to the received packet of data thereby forming an extended packet of data which includes both the packet header and the meta header, wherein the received packet of data is an isochronous packet of data received over an isochronous channel, and storing the extended packet of data onto a media within the media storage device. As described above, Kuver does not teach adding a meta header to the ***received*** packet of data, the received packet of data including a packet header, thereby forming an extended packet of data which includes both the packet header and the meta header and storing the extended packet of data onto a media within the media storage device. For at least these reasons, the independent Claim 44 is allowable over the teachings of Kuver.

The independent Claim 50 is directed to a method of writing data to a media storage device. The method of Claim 50 comprises receiving a received packet of data to be written to the media storage device, the received packet of data including a packet header and a common isochronous packet header, adding a meta data header to the received packet of data thereby forming an extended packet of data which includes the packet header, the common isochronous packet header and the meta data header and storing the extended packet of data onto a media within the media storage device. As described above, Kuver does not teach adding a meta data header to a ***received*** packet that includes a packet header and a common isochronous packet header to form an extended packet of data and storing the extended packet of data onto a media within a media storage device. For at least these reasons, the independent Claim 50 is allowable over the teachings of Kuver.

The independent Claim 51 is directed to a media storage device. The media storage device of Claim 51 comprises an interface circuit configured to receive a stream of data, thereby forming a received stream of data, and also to transmit a retrieved stream of data, storage media configured to store and retrieve the received stream of data, wherein the received stream of data includes one or more received packets of data, each including both a packet header and a common isochronous packet header, and an embedded stream processor coupled to the interface circuit and to the storage media to add a meta data header to each received packet in the received stream of data as it is received, thereby forming an extended packet of data, and provide the extended packet of data to the storage media for recording to form a recorded stream of data, the meta data header including a cycle mark value marking cycle boundaries within the recorded stream of data. As described above, Kuver does not teach receiving a stream of data including one or more received packets, each including both a packet header and a common isochronous packet header and adding a meta data header to each ***received*** packet in the received stream of data. For at least these reasons, the independent Claim 51 is allowable over the teachings of Kuver.

Rejections Under 35 U.S.C. § 103

Within the Office Action, Claims 1-13, 19-26, 29-32, 35 and 44-51 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Applicant Admitted Prior Art ("the AAPA") in view of U.S. Patent No. 6,012,117 to Traw et al. ("Traw"). The Applicants respectfully disagree with this rejection.

Referring to Figure 4A of the present invention, which is designated as prior art, and the accompanying description, the isochronous and CIP headers are added to the isochronous data packet before the packet is ***transmitted*** by the source device 50 over the IEEE 1394-1995 serial bus to the media storage device 40 of the present invention. Referring now to Figure 4B, which is in accordance with the present invention, not the prior art, a new header (the meta data header) is added by the media storage device 40 after the packet is ***received*** by the media storage device thereby forming an extended packet, and storing the extended packet on a media storage device. The extended packet includes both the packet header and the added meta data header. AAPA

does not teach or disclose adding a header after the packet is ***received*** by a media storage device thereby forming an extended packet of data which includes both the packet header and the added meta data header. Rather, AAPA simply teaches that the isochronous and CIP headers are inserted by the source device **prior to transmission** on the sending side. Specifically, AAPA states, "[o]nce the isochronous and CIP headers are applied to the isochronous data packet, the packet is **then transmitted** by the source device 50 over the IEEE 1394-1995 serial bus to the media storage device 40." [present specification, page 13, lines 4-6, emphasis added] The corresponding figure, Figure 4A, includes adding headers; however, as quoted above, the headers are added before the packet is transmitted, not after it is received. Again, within the Office Action, the concepts of transmission and reception are being fundamentally confused and misapplied. For AAPA to teach the present invention, the AAPA would have to teach CIP headers added to the isochronous data packet after the packet is received, not before the packet is transmitted. Hence, what Applicant admitted on page 12, lines 12 and 13, 8/6/04 does not disclose the present invention. The language that the Examiner quotes regarding page 12, lines 12 and 13, 8/6/04, highlights exactly the misunderstanding between transmission and reception discussed above. The language quoted in the Office Action response states, "CIP headers are added to the isochronous data packet **before** the packet is **transmitted**." Therefore, if the headers are added before transmission, they are not added after the packet is **received**. Hence, AAPA does not teach adding a meta data header to the **received** packet of data. The language of the present specification used for AAPA is very explicit regarding this, it has been quoted above and should not be confused. Clearly, the headers in AAPA are added before the packet is transmitted, thus not added to a **received** packet of data.

As recognized within the Office Action, Traw also does not teach adding a header to a received packet of data thereby forming an extended packet of data. Traw also does not teach storing the extended packet of data onto a media within the media storage device. Accordingly, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach adding a header to a ***received*** packet of data thereby forming an extended packet of data which includes both a packet header and a meta data header and storing the extended packet of data on to a media within the media storage device.

In contrast to the teachings of the specification of the present invention which is designated as prior art, Traw and their combination, the method of and apparatus for writing and reading time sensitive data within a storage device of the present invention receives a received packet of data to be written to the media storage device, adds a header to the received packet of data thereby forming an extended packet of data, and stores the extended packet of data onto a media within the media storage device. **The extended packet of data includes both the packet header and the meta data header.** Referring to Figure 4A of the present application, the data packet 80, prior to transmission by the source device 50, includes the Isoch header, the CIP header and the data blocks, as described previously. However, after ***receipt*** of the packet of data, the present invention adds the Meta-Data Header 82 (Figure 4B), in contrast to the configuration described in either the specification of the present invention which is designated as prior art, Traw, or their combination. As described above, neither the specification of the present invention which is designated as prior art, Traw, nor their combination, teach receiving a received packet of data to be written to the media storage device, adding a header to the ***received*** packet of data thereby forming an extended packet of data and storing the extended packet of data onto a media within the media storage device.

As clearly described above using quoted language, AAPA only teaches adding headers to a packet of data **before** the packet is **transmitted**, thus it does not teach adding a header to a **received** packet of data. Hence, AAPA does not teach or make obvious the present invention. It is recognized within the Office Action that Traw does not teach adding a header to a received packet of data. Hence, Traw does not teach or make obvious the present invention. Accordingly, neither AAPA, Traw nor their combination teach or make obvious the present invention.

The independent Claim 1 is directed to a method of writing data to a media storage device. The method of Claim 1 comprises receiving a received packet of data to be written to the media storage device, the received packet of data including a packet header, adding a meta data header to the received packet of data thereby forming an extended packet of data including both the packet header and the meta data header, and storing the extended packet of data onto a media within the media storage device. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach adding a meta data

header to the ***received*** packet of data which includes a packet header, thereby forming an extended packet of data which includes both the packet header and the meta data header, and storing the extended packet of data onto a media within the media storage device. For at least these reasons, the independent Claim 1 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Claims 2-7 are all dependent on the independent Claim 1. As discussed above, the independent Claim 1 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination. Accordingly, the dependent Claims 2-7 are all also allowable as being dependent on an allowable base claim.

The independent Claim 8 is directed to a method of reading data from a media storage device which has previously been stored with header data generated by the media storage device. The method of claim 8 comprises locating a first header data, including a cycle mark value having a pattern, reading a previously stored packet of data following the first header data from a media within the media storage device, the previously stored packet of data including a packet header, stripping the first header data from the previously stored packet of data thereby forming a retrieved packet of data, and transmitting the retrieved packet of data to another device. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach generating header data by a media storage device, stripping the first header data from the previously stored packet of data which includes a packet header, thereby forming a retrieved packet of data and transmitting the retrieved packet of data to another device. For at least these reasons, the independent Claim 8 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Claims 9-13 are all dependent on the independent Claim 8. As discussed above, the independent Claim 8 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination. Accordingly, the dependent Claims 9-13 are all also allowable as being dependent on an allowable base claim.

The independent Claim 19 is directed to a meta data header added to received packets by a media storage device as the packets are recorded on storage media within the media storage

device, each of the received packets including an existing header to which the meta data header is added such that the received packets include both an existing header and a meta data header. The meta data header of Claim 19 comprises a cycle mark value including a pattern used to locate cycle boundaries within the received packets and a cycle count value specifying a cycle number of a cycle in which the received packets are received. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach adding a meta data header to an existing header of ***received*** packets by a media storage device, a cycle mark value including a pattern used to locate cycle boundaries within the received packets and a cycle count value specifying a cycle number of a cycle in which the received packets are received. For at least these reasons, the independent Claim 19 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Claims 20-23 are all dependent on the independent Claim 19. As discussed above, the independent Claim 19 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination. Accordingly, the dependent Claims 20-23 are all also allowable as being dependent on an allowable base claim.

The independent Claim 24 is directed to a media storage device. The media storage device of Claim 24 comprises means for interfacing configured for receiving a stream of data, thereby forming a received stream of data, and also for transmitting a retrieved stream of data, the received stream of data including packet header data, means for storing data for storing and retrieving the received stream of data, and means for processing coupled to the means for interfacing and to the means for storing for adding meta header data to the received stream of data as the received stream of data is received, such that each packet within the received stream of data includes both packet header data and meta header data, and providing the meta header data and the received stream of data to the means for storing for recording thereby forming a recorded stream of data, the meta header data including a cycle mark value marking cycle boundaries within the recorded stream of data. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach a means for processing for adding meta header data to the ***received*** stream of data which includes packet

header data, as the received stream of data is received and providing the meta header data and the received stream of data to the means for storing for recording thereby forming a recorded stream of data. For at least these reasons, the independent Claim 24 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Claims 25, 26 and 29 are all dependent on the independent Claim 24. As discussed above, the independent Claim 24 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination. Accordingly, the dependent Claims 25, 26 and 29 are all also allowable as being dependent on an allowable base claim.

The independent Claim 30 is directed to a media storage device. The media storage device of Claim 30 comprises an interface circuit configured to receive a stream of data, thereby forming a received stream of data, and also to transmit a retrieved stream of data, the received stream of data including packet header data, storage media configured to store and retrieve the received stream of data, and an embedded stream processor coupled to the interface circuit and to the storage media to add meta header data to the received stream of data as it is received, such that each packet within the received stream of data includes both packet header data and meta header data, and provide the meta header data and the received stream of data to the storage media for recording to form a recorded stream of data, the meta header data including a cycle mark value marking cycle boundaries within the recorded stream of data. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach an embedded stream processor to add meta header data to the ***received*** stream of data which includes packet header data, as it is received and provide the meta header data and the received stream of data to the storage media for recording to form a recorded stream of data. For at least these reasons, the independent Claim 30 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Claims 31, 32 and 35 are all dependent on the independent Claim 30. As discussed above, the independent Claim 30 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination. Accordingly, the dependent Claims 31, 32 and 35 are all also allowable as being dependent on an allowable base claim.

The independent Claim 44 is directed to a method of writing data to a media storage device. The method of Claim 44 comprises receiving a received packet of data to be written to the media storage device, the received packet of data including a packet header, adding a meta header to the received packet of data thereby forming an extended packet of data which includes both the packet header and the meta header, wherein the received packet of data is an isochronous packet of data received over an isochronous channel, and storing the extended packet of data onto a media within the media storage device. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach adding a meta header to the ***received*** packet of data which includes a packet header, thereby forming an extended packet of data which includes both the packet header and the meta header and storing the extended packet of data onto a media within the media storage device. For at least these reasons, the independent Claim 44 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Claims 45-49 are all dependent on the independent Claim 44. As discussed above, the independent Claim 44 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination. Accordingly, the dependent Claims 45-49 are all also allowable as being dependent on an allowable base claim.

The independent Claim 50 is directed to a method of writing data to a media storage device. The method of Claim 50 comprises receiving a received packet of data to be written to the media storage device, the received packet of data including a packet header and a common isochronous packet header, adding a meta data header to the received packet of data thereby forming an extended packet of data which includes the packet header, the common isochronous packet header and the meta data header and storing the extended packet of data onto a media

within the media storage device. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach adding a meta data header to a ***received*** packet that includes a packet header and a common isochronous packet header to form an extended packet of data which includes the packet header, the common isochronous packet header and the meta data header and storing the extended packet of data onto a media within a media storage device. For at least these reasons, the independent Claim 50 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

The independent Claim 51 is directed to a media storage device. The media storage device of Claim 51 comprises an interface circuit configured to receive a stream of data, thereby forming a received stream of data, and also to transmit a retrieved stream of data, storage media configured to store and retrieve the received stream of data, wherein the received stream of data includes one or more received packets of data, each including both a packet header and a common isochronous packet header, and an embedded stream processor coupled to the interface circuit and to the storage media to add a meta data header to each received packet in the received stream of data as it is received, thereby forming an extended packet of data, and provide the extended packet of data to the storage media for recording to form a recorded stream of data, the meta data header including a cycle mark value marking cycle boundaries within the recorded stream of data. As described above, neither the specification of the present invention which is designated as prior art, Traw nor their combination teach receiving a stream of data each including a packet header and adding a meta data header to each ***received*** packet in the received stream of data. For at least these reasons, the independent Claim 51 is allowable over the teachings of the specification of the present invention which is designated as prior art, Traw and their combination.

Within the Office Action, Claims 14, 15, 27, 28, 33 and 34 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over the AAPA in view of Traw as applied to Claim 1 above, and further in view of Kuver. Claims 14 and 15 are dependent on the independent Claim 8. Claims 27 and 28 are dependent on the independent Claim 24. Claims 33 and 34 are dependent on the independent Claim 30. As discussed above, the independent Claims 8, 24 and

30 are all allowable over the teachings of the AAPA, Traw and their combination. Accordingly, the dependent Claims 14, 15, 27, 28, 33 and 34 are all also allowable as being dependent on an allowable base claim.

For the reasons given above, Applicants respectfully submit that the claims are now in a condition for allowance, and allowance at an early date would be appreciated. Should the Examiner have any questions or comments, they are encouraged to call the undersigned at (408) 530-9700 to discuss the same so that any outstanding issues can be expeditiously resolved.

Respectfully submitted,
HAVERSTOCK & OWENS LLP

Dated: January 20, 2006

By: Jonathan O. Owens
Jonathan O. Owens
Reg. No. 37,902
Attorneys for Applicant(s)

CERTIFICATE OF MAILING (37 CFR 1.8a)

I hereby certify that this paper (along with any referenced being attached or enclosed) is being deposited with the U.S. Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to the: Commissioner for Patents, P.O. Box 1450 Alexandria, VA 22313-1450

HAVERSTOCK & OWENS LLP.

Date: 1-20-06 By: